Sabanci University

Faculty of Engineering and Natural Sciences

CS 308 Software Engineering
Online Store Project

As a course project, you will be developing a website for an online store, which interacts over the web with an application web-server. The choice of technology is up to the project team.

The basic requirements for the system are given as follows:

1. The application shall present a number of products in categories and let users select and add the desired product/products to the shopping cart to purchase them.

2. You are free to design your store for any kind of product that your team prefers. For instance, it can be an electronics appliance store such as *teknosa.com* or clothing store of a brand and so on.

3. The store has a limited stock of items, when the user selects a product, the number of items in the stock must be shown. When the shopping is done, that product should be decreased from the stock and the order for delivery processing should be forwarded to the delivery department, which will process the order for shipping. During order processing the user should be able to see the status as: *processing*, *in-transit*, and *delivered*. **(10%)**

4. Users can browse the online store and add products to their shopping carts without logging in to the system. They, however, should login before placing an order and making a payment. Once payment is made and confirmed by the (mock-up) banking entity, an invoice must be shown on the screen and a pdf copy of the invoice should be emailed to the user. **(10%)**

5. Users should be able to make comments and give ratings to the products. The ratings typically are between 1 and 5 stars or 1 and 10 points. The comments should be approved by the product manager before they become visible. **(10%)**

6. The application should have an attractive, easy-to-use, and professional-looking graphical user interface **(5%).**

7. The user should be able to search products depending on their names or descriptions. Additionally, the user should be able to sort products depending on their price or popularity. If a product is out-of-stock the product must still be searchable, however the user should not be able to add it to the shopping cart. **(10%)**

8. The user should be able to browse and purchase the products through the website. Additionally, the website should provide an admin interface for managerial tasks. **(10%)**

9. A product should have the following properties at the very least: ID, name, model, serial number, description, quantity in stocks, price, warranty status, and distributor information.

10. There are three types of basic roles in this system; customers, sales managers, and product managers.

11. The sales managers are responsible for setting the prices of the products. They shall set a discount on the selected items. When the discount rate and the products are given, the system automatically sets the new price and notify the users, whose wish list includes the discounted product, about the discount. They shall also view all the invoices in a given date range, can print them or save them as "pdf" files. Last but not least, they shall calculate the revenue and loss/profit in between given dates and view a chart of it. **(10%)**

12. The product managers shall add/remove products as well as product categories, and manage the stocks. Everything related to stock shall be done by the product manager. The product manager is also in the role of delivery department since it controls the stock. This means, the product manager shall view the invoices, products to be delivered, and the corresponding addresses for delivery. A delivery list has the following properties: delivery ID, customer ID, product ID, quantity, total price, delivery address, and a field showing whether the delivery has been completed or not. Last but not least, the product managers shall approve or disapprove the comments. **(10%)**

13. The customers shall view the products, search for the products, comment on the products, rate the products, include products in their wishlists, place new orders, cancel existing orders, and return previously purchased products. A customer has the following properties at the very least: ID, name, tax ID, e-mail address, home address, and password. **(10%)**

14. A customer should enter his/her credit card information to purchase a product. Credit card verification and limit issues are out of scope of the project. **(3%)**

15. A customer shall also be able to selectively return a product and ask for a refund. In such a case, the customer will select an already purchased product from his/her order history within 30 days of purchase. The sales manager will evaluate the refund request and upon receiving the product back to the store will authorize the refund. The product will be added back to the stock and the purchased price will be refunded to the customer's account. Moreover, if the product was bought during a discount campaign and the customer chooses to return the product after the campaign ends, the refunded amount will be the same as the time of its purchase with the discount applied. **(10%)**

16. Since the registration and payment process is sensitive in nature; your project development and programming should reflect the necessary amount of security aware-ONLINE-ness and defensive programming. The various user roles have their own security privileges and they should not be mixed. Whatever your method of information

storage (databases, XML files etc.) is, sensitive information should be kept encrypted, so that it's not easily compromised. Note that sensitive information includes the following at the very least: user passwords, credit card information, the invoices, and the user accounts. **(1%)**

17. Needless to say, your software is expected to run smoothly and not to display any unexpected behavior while functioning within its normal parameters. Additionally, since this system will serve a potentially large number of users, you should keep concurrency in mind: Your system should be able to handle multiple users of various roles working on it at the same time and retain its normal functionality under such circumstances. **(1%)**

    *be aware that, **(X%)** for the listed features above indicates their contribution in the grading*

Project Schedule

Date	Project Process
15.10.2024	Sprint 1 Planning and Sprint 1 Starts
22.10.2024	Sprint 2 Planning
29.10.2024 (will be rescheduled due to Republic Day)	Sprint 1 Review & Retrospective and Sprint 2 Starts
05.11.2024	Sprint 3 Planning
12.11.2024	Sprint 2 Review & Retrospective and Sprint 3 Starts
19.11.2024	Sprint 4 Planning
03.12.2024	**Progress Demo**
10.12.2024	Sprint 3 Review & Retrospective and Sprint 4 Starts
17.12.2024	Sprint 5 Planning
24.12.2024	Sprint 4 Review & Retrospective and Sprint 5 Starts
31.12.2024 (will be rescheduled due to New Year's Eve)	Sprint 5 Review & Retrospective
TBA	**Final Demos**

The table given above presents the project schedule. Note that we are using 2-week sprints, which indicates that we expect to see a new release of your systems every other week. Furthermore, the sprints are overlapping in the sense that in the middle of a sprint we hold the sprint planning meeting to plan for the subsequent sprint. All the SCRUM meetings indicated above (i.e., the sprint planning, review, and retrospective meetings) will be held during the lab

hours. To this end, the TAs will schedule 30-minute online meetings for each team that they are responsible for. For the weeks where we have the sprint planning meetings, 30 minutes will be spent on the planning. For the weeks where we have the sprint review and retrospective meetings, the TA will join the last 15 minutes of the review meetings and the first 15 minutes of the retrospective meetings. Therefore, the teams should meet before their scheduled meeting times with the TAs (such that the TAs can join the meeting later) and stay in the meetings even after the TAs leave.

Note further that you will be demoing your project twice; once during the semester as a progress demo and once typically during and/or after the final exams as the final demo. You will be graded for both of them (see below for the grading criteria).

For **Progress Demo**, we will be expecting at least the following numbered features **1, 3, 4, 5, 7** and **9** from the above requirements to be presented. However, for the final demo all of the features listed above will be expected.

At the end of the semester, you are expected to deliver a professional-looking, ready-to-use product. Therefore you should learn/use the optimal frameworks for the implementation. You are free to choose any framework, which you feel comfortable with. We, however, encourage you to consider frameworks that are well-documented, align with the project requirements, and are widely recognized for promoting the best practices in software development.

Also be aware that, in our experience, occasionally some teams experience that some of their members don't fulfill their assigned responsibilities and don't contribute as expected, which leads to negatively affecting the performance of the team. If your team experiences this or similar issues, it is the responsibility of the other team members to first communicate with those members in order to rectify the situation. However, if this does not work, the team should immediately inform the assigned TA and the instructor about the situation, so that corrective action(s) can be taken.

Technologies and Frameworks

Below is the list of suggested technologies that you may want to use in your projects. Note that this list is by no means comprehensive and you are free to choose other technologies as you sit fit.

Backend	
	NodeJS
	Django
	Laravel
	RubyOnRails
	Flask
	FastApi
	SpringBoot

Database	
	Mysql
	PostgreSQL
	Mongodb
	Firebase
	SQLite

Frontend Web	
	Angular
	ReactJS
	Bootstrap
	JQuery & JQueryUI
	Flutter
	VueJS

General Advices

Here are some general advices:

1. You can serve the website directly through the framework or provide a web API for frontend-backend interaction. Nonetheless, the chosen backend (server-side) will connect with a database for data storage, handle concurrent http requests and serve web based resources.

2. We strongly suggest you familiarize yourself with the different tools/frameworks that can be used in your projects before starting the implementation. Make sure that you and your team are ready and well versed for their assigned tasks. To decide on the tools/frameworks to be used, take into account the experience of your team members as well as the usability, the documentation, and the popularity of the framework. Keep in mind that changing a backend server-side framework during the development process will be costly!

3. Teams must ensure that each team member is consistently contributing to the project. If, for any reason, you feel that this is not the case, inform the respective TA and the course instructor immediately, so that necessary precautions can be taken in time.

4. Keep in mind the milestones and the amount of resources you have for the project, so that you can plan your development efforts accordingly. We generally suggest that you stay ahead or in-sync with the schedule given above. Note that falling behind the specified schedule, typically contributes to the technical debt you owe, which (especially for short projects) can accumulate quite fast, resulting in failed projects. You should especially keep an eye on the demo dates and work accordingly.

5. If you happen to use the Python-based *Django* framework, please refrain from developing your admin panel of the project with the built-in admin interface provided by this framework as this doesn't require any development effort. We expect you to develop your own admin interface, as this carries a portion of the project grade. If you violate this constraint, you may not receive any points for the admin interface implementation.

Project Grading

The following tables present the **tentative** project grading scheme:

Project Grading	Points (overall effect in the final course grade)
Progress Demo	20%
Final Demo	30%

For each demo, the grading will comprise on two parts:
- 80% from your demo presentation. The percentage points for each feature are given in the feature list above. For the progress demo, the percentages of the requested features will be used as weights, such that the total adds up to 100%.
- 20% from your development activities, which are further distributed as:

Demo Grading	Points (20% of the demo grade)
At least 5 commits per member per demo	18%
At least 25 new unit test cases per demo	18%
At least 5 new bug reports per demo	18%
At least 15 product backlog items per demo	18%
At least 30 sprint backlog items per demo	18%
Attendance in the SCRUM meetings	10%